GENESIS - Learning Outcome & Mini-project Summary Report

Details

Ver. Rel. No.	Release Date	Prepared By	Module Name	To Be Approved	Remarks/Revision Details
1.0	16/02/2022	Midhun Chakravarthi 40020506	C Programming On Multiple Platforms		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Essesntials of Embedded Systems		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Applied SDLC and Software Testing		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Applied Model Based Design Module		
1.0	16/02/2022	Midhun Chakravarthi 40020506	OOPS with Python		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Mastering Microcontrollers with Embedded Driver Development Module		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Overview of Automotive Systems		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Applied Control Systems and Vehicle Dynamics		
1.0	16/02/2022	Midhun Chakravarthi 40020506	Classic Autosar Basic to Intermediate		

Contents

List of Figures

Miniproject – 1: Tic Tac Toe [Individual]

Modules:

1. C Programming
2. Git

Requirements

4W's and 1 H's

Why:

This classic game contribute to the children's development growth in numerous ways include their understanding of predictability, problem solving, spatial reasoning, hand eye coordination, turn taking and stratezing.

Where:

1. It can be used in our daily lives.

Who:

1. It can be used by anyone(EX: students,teachers etc.).
2. Can be used to relief stress.

What:

Tic-tac is a game for two players who take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner.

How:

The game is to be played between two people .

One of the player chooses ‘O’ and the other ‘X’ to mark their respective cells.

The game starts with one of the players and the game ends when one of the players has one whole row/ column/ diagonal filled with his/her respective character (‘O’ or ‘X’).

If no one wins, then the game is said to be draw.

High Level Requirements

HLR	DESCRIPTION
HLR_1	User shall be able to choose 'X' to play
HLR_2	User shall be able to choose 'O' to play
HLR_3	User shall be able to Exit the game
HLR_4	User shall lose
HLR_4	User shall win
HLR_5	User shall end up in a Draw situation

Low Level Requirements

LLR	DESCRIPTION	HLR_ID
LLR_1	If the user presses '1', he'll be play with 'X'.	HR01
LLR_2	If the user is playing with 'X', he'll get the first turn.	HR01
LLR_3	If the user presses '2', he'll be play with 'O'.	HR02
LLR_4	If the user is playing with 'O', he'll get the second turn.	HR02
LLR_5	If the user presses '3', it'll exit the game.	HR03
LLR_6	If the computer gets 3 Xs or 3 Os in vertical,horizontal or diagonal row, User will lose.	HR03
LLR_7	If the user gets 3 Xs or 3 Os(as per his choice), in vertical,horizontal or diagonal row, he'll win.	HR04
LLR_8	If the total number of moves, i.e., 9 moves have been completed and neithe the user nor the computer has won, it'll end up in a draw.	HR04

Design

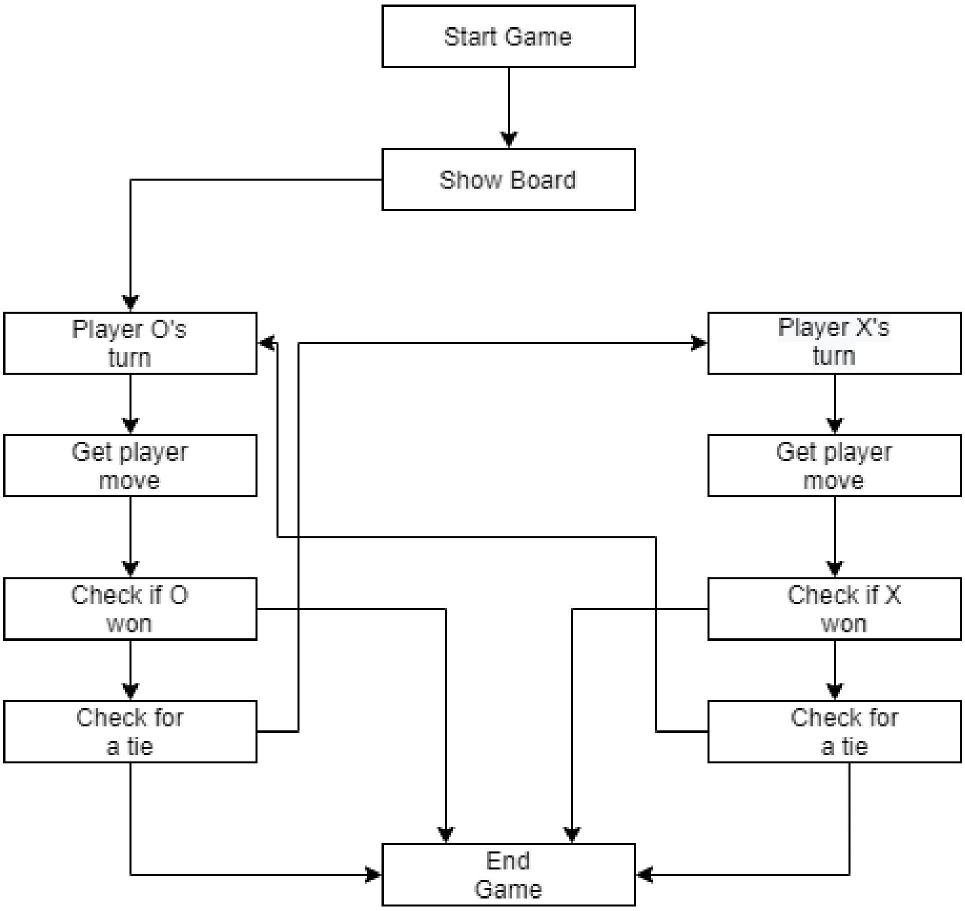


Figure 1 Behavior Diagram

Test Plan

High Level Test Plan

Test ID	Description	Exp I/P	Exp o/p
H_01	Check if the graph for playing is being drawn or not.	No input.	3X3 graph is drawn.
H_02	Check if player/computer got 3 of his inputs in vertical, horizontal or diagonal format.	'X' or 'O' i/p from the user/computer.	The user/computer won the game.
H_03	Check for draw.	9 inputs from (user+computer).	The game is over.

Low Level Test Plan

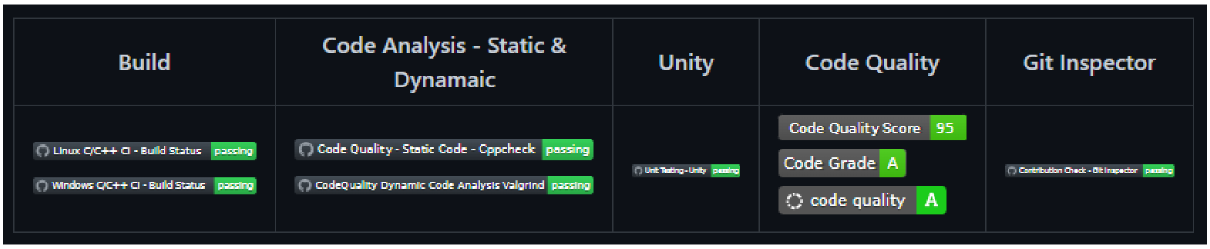
Test ID	Description	Exp I/P	Exp o/p
L_01	Checking for the basic requirement to the game, i.e., a 3X3 graph is drawn or not. This 3X3 graph is the basic need to play the game as it is like a game board for the game.	Not input expected from the user.	3X3 graph is drawn.
L_02	Play proceeds with the user/computer alternately placing their marks in any unoccupied cell. Check if any player/computer finishes with 3 marks in a row(vertical, horizontal or diagonal).	'X' or 'O' i/p from the user/computer.	The user/computer won the game.
L_03	Check if a total of 9 moves have been made(combining that of user and computer), the game ends up in a draw when neither the user nor the computer is able to get 3 marks in a row.	9 inputs from (user+computer).	The game is over. Somebody won or the game ended as a draw.

Implementation and Summary

Git Link:

Link: https://github.com/MidhunChakravarthi-06/M1_application_SudokuGame

Git Dashboard



Build	Code Analysis - Static & Dynamaic	Unity	Code Quality	Git Inspector
Linux C/C++ CI - Build Status passing Windows C/C++ CI - Build Status passing	Code Quality - Static Code - Cppcheck passing CodeQuality Dynamic Code Analysis Valgrind passing	Unit Testing - Unity passing	Code Quality Score 95 Code Grade A code quality A	Contribution Check - GitInspector passing

Figure 2 Git Dashboard

Miniproject 2 –Seat Heating System [Individual]

Modules

1. C Programming
2. Embedded System
3. SimulIDE
4. Git

Requirements

Heated seats are powered by a heating element, a long strip of material that functions as a resistor. A resistor resists the flow of electricity. When electric current flows through it, the energy is turned into heat, which flows through the seat.

4W's and 1 H's

Why:

1. To maintain the temperature inside the car for not to catch cold.
2. To warm quickly from outside of the car.

Where:

1. This can be used in our cars.

Who:

1. Can be used by the drivers of the car

When:

1. When outerside of the car is too cold
2. This project is used to prevent people from high cold.

How:

1. It sense the temperature by sensor and heat the seat through resistor.

Design

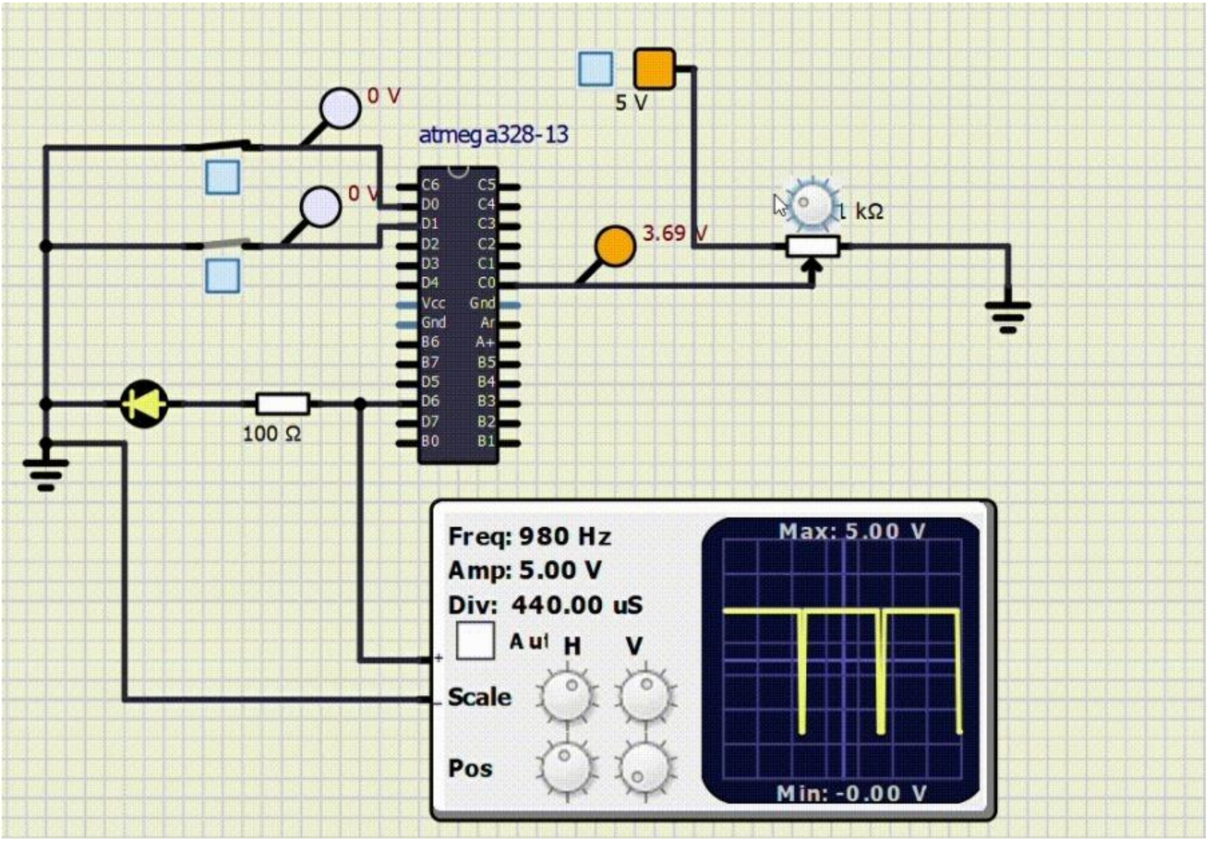


Figure 3 Sample diagram

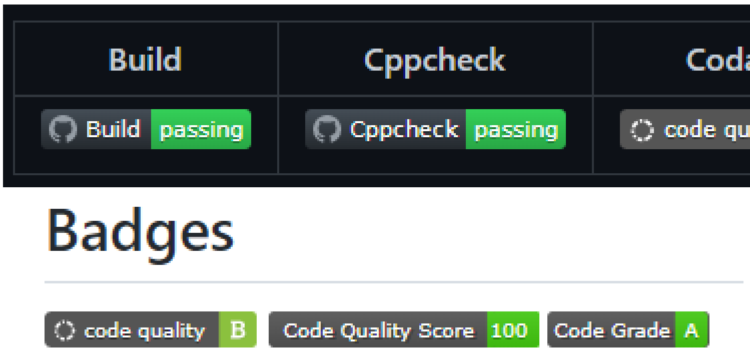
Test Plan

Implementation and Summary

Git Link:

Link: https://github.com/MidhunChakravarthi-06/M2-Embedded_Smartlock

Git Dashboard



Build	Cppcheck	Codacy
Build passing	Cppcheck passing	code quality A

Badges

code quality B | Code Quality Score 100 | Code Grade A

Figure 4 Git Dashboard

Miniproject 3 – Chatter Bot [Team]

Modules

1. SDLC
2. Git

Requirements

4W's and 1 H's

Why:

1. Software application used to conduct an online chat conversation via text or speech.
2. A Computer program which simulates a natural human conversation.

Where:

1. Retail and E-Commerce industries.
2. Used in Healthcare.

Who

1. Clients who need assistance.
2. Peoples who need support.

When:

1. To Provide faster and cheaper assistance to client.
2. To be Increasingly comfortable with Technology.

How:

1. Customers who are dealing with their problems late at night, chatbot are blessing as they can work around the clock.
2. During conversations with the customers, chat box provides a bridge between sales and customer team.

High Level Requirements

ID	Description	Category	Status
HLR_1	Chatterbot	Technical	Implemented

Low Level Requirements

ID	Description	HLR ID	Status
LLR_1	Process input	HLR_1	Implemented
LLR_2	Logic adapter 1	HLR_1	Implemented
LLR_3	Logic adapter 2	HLR_1	Implemented

Design

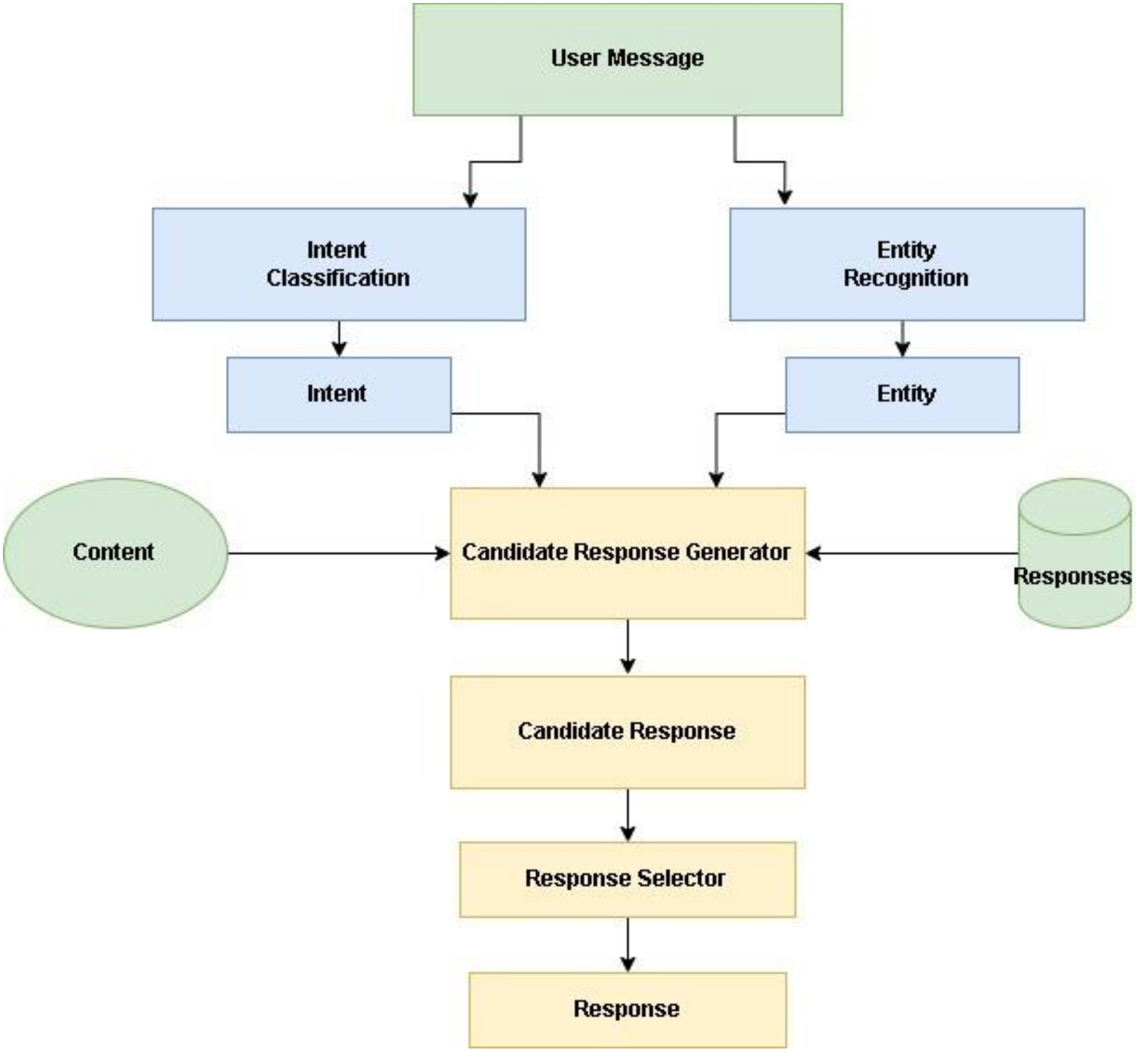


Figure 5Behaviour diagram

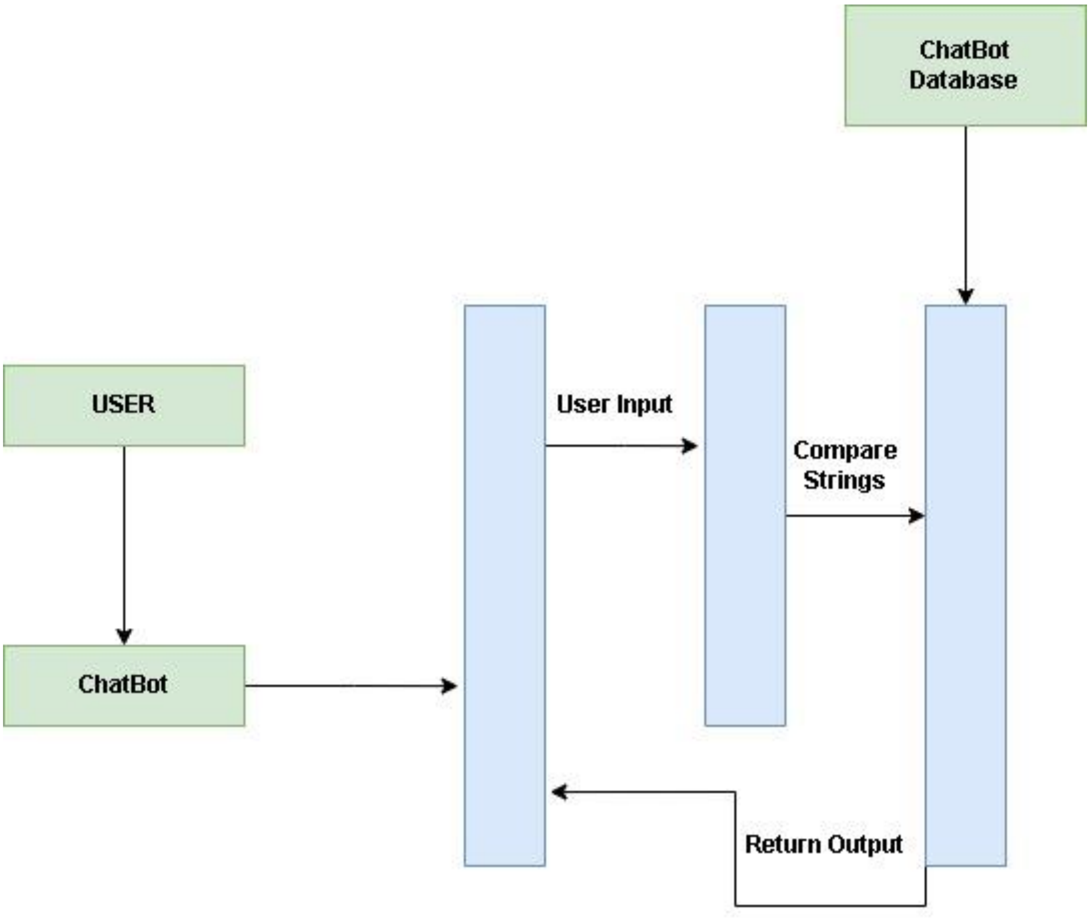


Figure 6 Userflow diagram

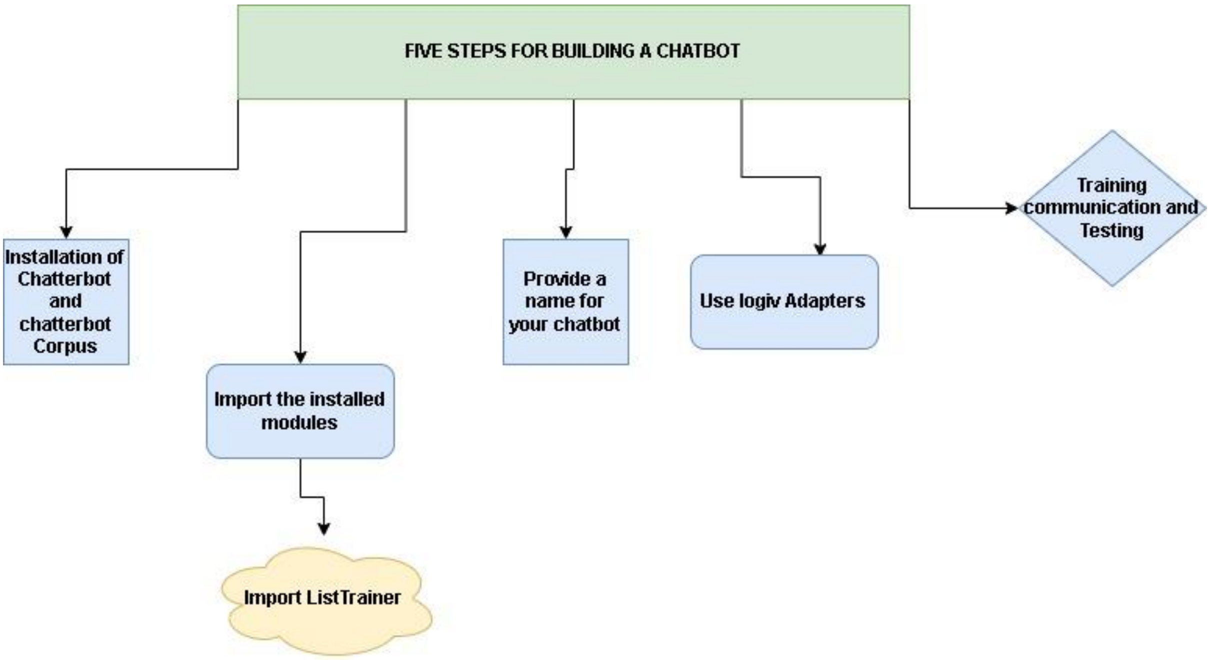

FIVE STEPS FOR BUILDING A CHATBOT
Installation of Chatterbot and chatterbot Corpus
Import the installed modules
Import ListTrainer
Provide a name for your chatbot
Use logiv Adapters
Training communication and Testing

Figure 9 Structure Diagram

Test Plan

High Level Test Plan

Test ID	Description	Exp I/P	Exp O/P	Actual Out	Type Of Test
HLTP_1	Get input	User input	Return user input to the Process input	SUCCESS	Requirement Based
HLTP_2	Read input	Process input	Return a response related to the given User input	SUCCESS	Requirement Based
HLTP_3	Get output	Process input	Return the response from the Process input to the user	SUCCESS	Requirement Based

Low Level Test Plan

Test ID	HLTP ID	Description	Exp IN	Exp OUT	Actual Out	Type Of Test
LLTP_1	HLTP_1	The inputs can be given only by using console, API, speech recognition, etc.	User input	SUCCESS	SUCCESS	Requirement Based
LLTP_2	HLTP_2	Select a known statement that most closely matches the given User input	Process input	SUCCESS	SUCCESS	Requirement Based
LLTP_2.1	HLTP_2	Return a known response to	Process input	SUCCESS	SUCCESS	Requirement Based

Test ID	HLTP ID	Description	Exp IN	Exp OUT	Actual Out	Type Of Test
		the selected match and a confidence value based on the matching				
LLTP_3	HLTP_3	Return the response to the user only by using console, API, speech recognition, etc.	User input	SUCCESS	SUCCESS	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/GENESIS2021Q1/Applied_SDLC-Dec_Team_1

Individual Contribution and Highlights

Summary

1. Implementation
2. Testing

Role in Project Team

1. Implementation: Implemented a python code for test file.
2. Testing: Tested the Chatter Bot using spell checking.

Miniproject 4 – Attendance Automation[Team]

Modules

1. Python
2. Git

Requirements

High Level Requirements

ID	Description	Status
HLR_1	Attendance Status	Implemented
HLR_2	User Details	Implemented
HLR_3	User load Sheet	Implemented
HLR_4	Output File Generation	Implemented

Low Level Requirements

ID	Description	HLR ID	Status
LLR_1	User can get the attendance status	HLR_1	Implemented
LLR_2	User can enter status input to get the attendance status	HLR_1	Implemented
LLR_3	User can get the user details	HLR_2	Implemented
LLR_4	User will get the details after the successful attendance	HLR_2	Implemented
LLR_5	User can load different sheets	HLR_3	Implemented
LLR_6	User can modify the existing sheets as it is dynamic	HLR_3	Implemented
LLR_7	Output file gets generated	HLR_4	Implemented

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_1	Attendance Status	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_2	User details	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_3	User load sheet	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_4	Output file generation	User Input	SUCCESS	SUCCESS	Requirement Based

Low Level Test Plan

ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
LLTP_1	HLTP_1	User can get Attendance Status	SUCCESS	SUCCESS	Requirement Based
LLTP_2	HLTP_1	User can enter Status input to get the Attendance Status	SUCCESS	SUCCESS	Requirement Based
LLTP_3	HLTP_2	User can get the User details	SUCCESS	SUCCESS	Requirement Based
LLTP_4	HLTP_2	User will get the details after the successful attendance	SUCCESS	SUCCESS	Requirement Based
LLTP_5	HLTP_3	User can load different sheets	SUCCESS	SUCCESS	Requirement Based
LLTP_6	HLTP_3	User can also modify the existing sheets as it is dynamic	SUCCESS	SUCCESS	Requirement Based
LLTP_7	HLTP_4	Output file gets	SUCCESS	SUCCESS	Requirement

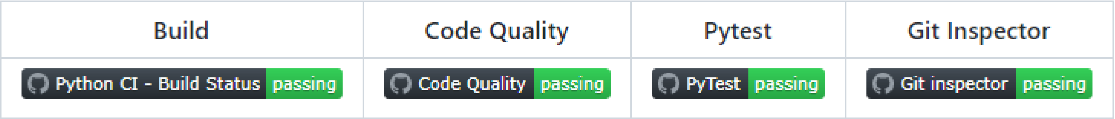
ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
		generated			Based

Implementation and Summary

Git Link:

Link: https://github.com/kavinvignes/GENESIS2021-OOPS_Python-Attendance_Automation-Team_13

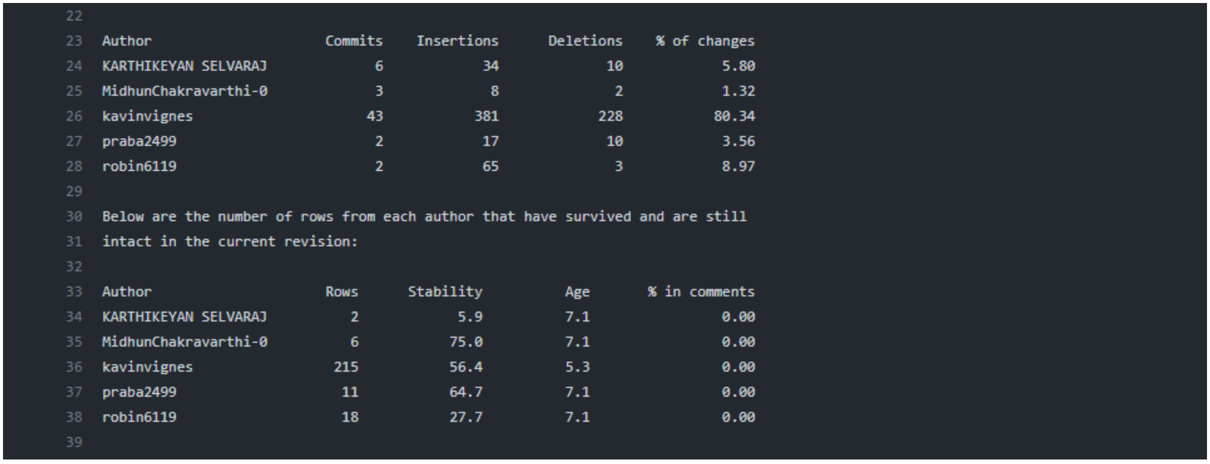
Git Dashboard



Build	Code Quality	Pytest	Git Inspector
Python CI - Build Status passing	Code Quality passing	PyTest passing	Git inspector passing

Figure 7 Git Dashboard

Git Inspector Summary



```
Author                 Commits    Insertions      Deletions    % of changes
KARTHIKEYAN SELVARAJ         6            34             10            5.80
MidhunChakravarthi-0         3             8              2            1.32
kavinvignes                 43           381            228           80.34
praba2499                    2            17             10            3.56
robin6119                    2            65              3            8.97

Below are the number of rows from each author that have survived and are still
intact in the current revision:

Author                     Rows      Stability          Age       % in comments
KARTHIKEYAN SELVARAJ          2            5.9          7.1                0.00
MidhunChakravarthi-0          6           75.0          7.1                0.00
kavinvignes                 215           56.4          5.3                0.00
praba2499                    11           64.7          7.1                0.00
robin6119                    18           27.7          7.1                0.00
```

Figure 8 Git Inspector Summary

Individual Contribution and Highlights

1. Improved implementation of Python Programming
2. Source code management using GitHub

Role in Project Team

1. Programmer: Done Programming for Attendance Automation
2. Tester: Writing Testcases for the main program.

Miniproject 5 – Tesla Project[Team]

Modules

1. Matlab
2. Git

Requirements

We have implemented following features

1. Autolock Control Door
2. Battery Monitoring System
3. Discharge Control
4. State Of Charge
5. Temperature Control System
6. Voltage Control
7. Warning system

Temperature Control System

The Battery Thermal Management System (BTMS) is the device responsible for managing/dissipating the heat generated during the electrochemical processes occurring in cells, allowing the battery to operate safely and efficiently.

The lithium-ion battery in electric vehicles is an important energy storage device that requires proper temperarature control system. A considerable amount of heat is generated by the battery cells owing to their internal resistance during charging and discharging, especially for peak vehicle loads.

This focus on developing a smart controlled temperature control solution in which a vapor compression system is integrated. A lumped-parameter cylindrical battery thermal model is developed with a Kalman observer to estimatethe transient changes in the temperatures of the battery surface, the battery core, and the cooling air flowing around the cells.

The optimal cooling airtemperature of the battery is investigated using optimal control theory. A model predictive controller is then introduced to regulate the refrigerant compressor and to track theideal cooling air temperature. The power consumption of the thermal management system and the behavior of the internal temperature.

Design

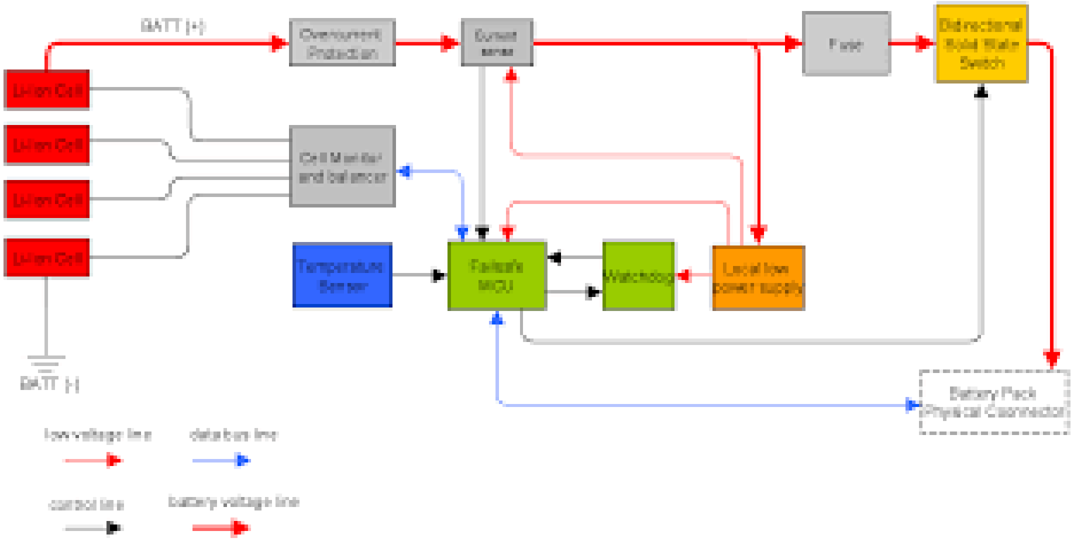


Figure 9 Block Diagram

Individual Contribution and Highlights

1. Implementation of Temperature Control System in Matlab simulation

Github Implementation Summary

Github Link: https://github.com/sunilkora31/TeamTesla_Applied_MBD.git

Miniproject 6 – Wiper Control[Team]

Modules

1. C Programming
2. STM32

Introduction

In this project, Wiper Control System is designed using STM32 Microcontroller.The STM32 family of 32-bit microcontrollers based on the Arm Cortex -M processor is designed to offer new degrees of freedom to MCU users. It offers products combining very high performance, real-time capabilities, digital signal processing, low-power / low-voltage operation, and connectivity, while maintaining full integration and ease of development.

Objective

The main objective of the system is to create a wiper control system with low power consumption and high performance, which can be achieved by STM 32 Microcontroller.

Components used

1. STM 32
2. Four LED
3. Switch

Research

The STM32 family of 32-bit microcontrollers based on the Arm Cortex M processor is designed to offer new degrees of freedom to MCU users. It offers products combining very high performance, real-time capabilities, digital signal processing, low-power / low-voltage operation, and connectivity, while maintaining full integration and ease of development. The unparalleled range of STM32 microcontrollers, based on an industry-standard core, comes with a vast choice of tools and software to support project development, making this family of products ideal for both small projects and end-to-end platforms.

Features

1. Low power Consumption
2. High performance
3. real-time capabilities

4W's and 1 H's

Why

1. To understand basic concepts in STM32
2. To control wiper system by switching LED in STM32

Where

1. It can be used for many mini projects.

2. Its has too many realistic features in this STM32 microcontroller

Who

1. It can be used by students.

2. 2.It can be used by anyone who are new to embedded programming language.

When

1. It can be used for both small projects and end-to-end platforms.

2. It is easy to access the emmbedded programing language.

How

1. By using softwares to exceute the program.

2. By loading the program in STM32 and execute.

High Level Requirements

Id	Description	Status
HLR_1	Microcontroller	Implemented
HlR_2	Swtich	Implemented
HLR_3	Four LED	Implemented
HLR_4	Software	Implemented

Low Level Requirements

Id	Description	Status
LLR_1	STM32	Implemented
LLR_2	Swtich	Implemented
LLR_3	Four LED	Implemented

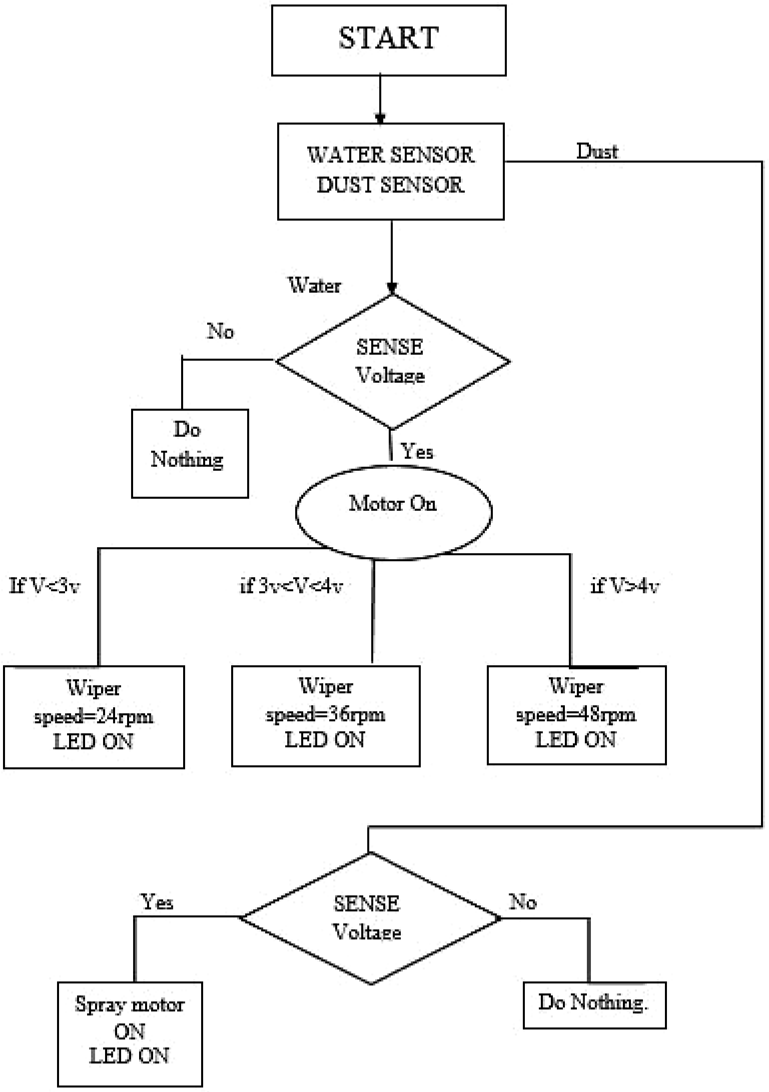


Figure 10 Structure Diagram

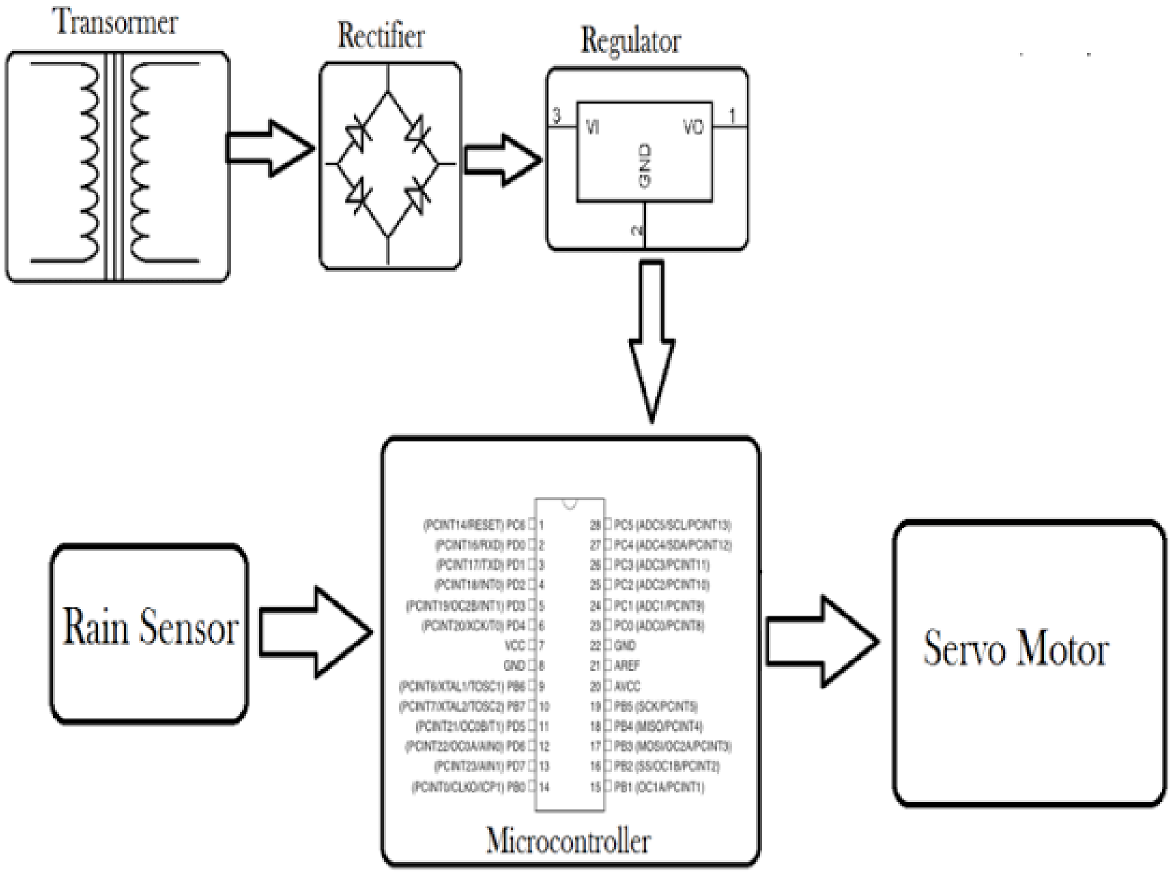


Figure 11 Behavior Diagram

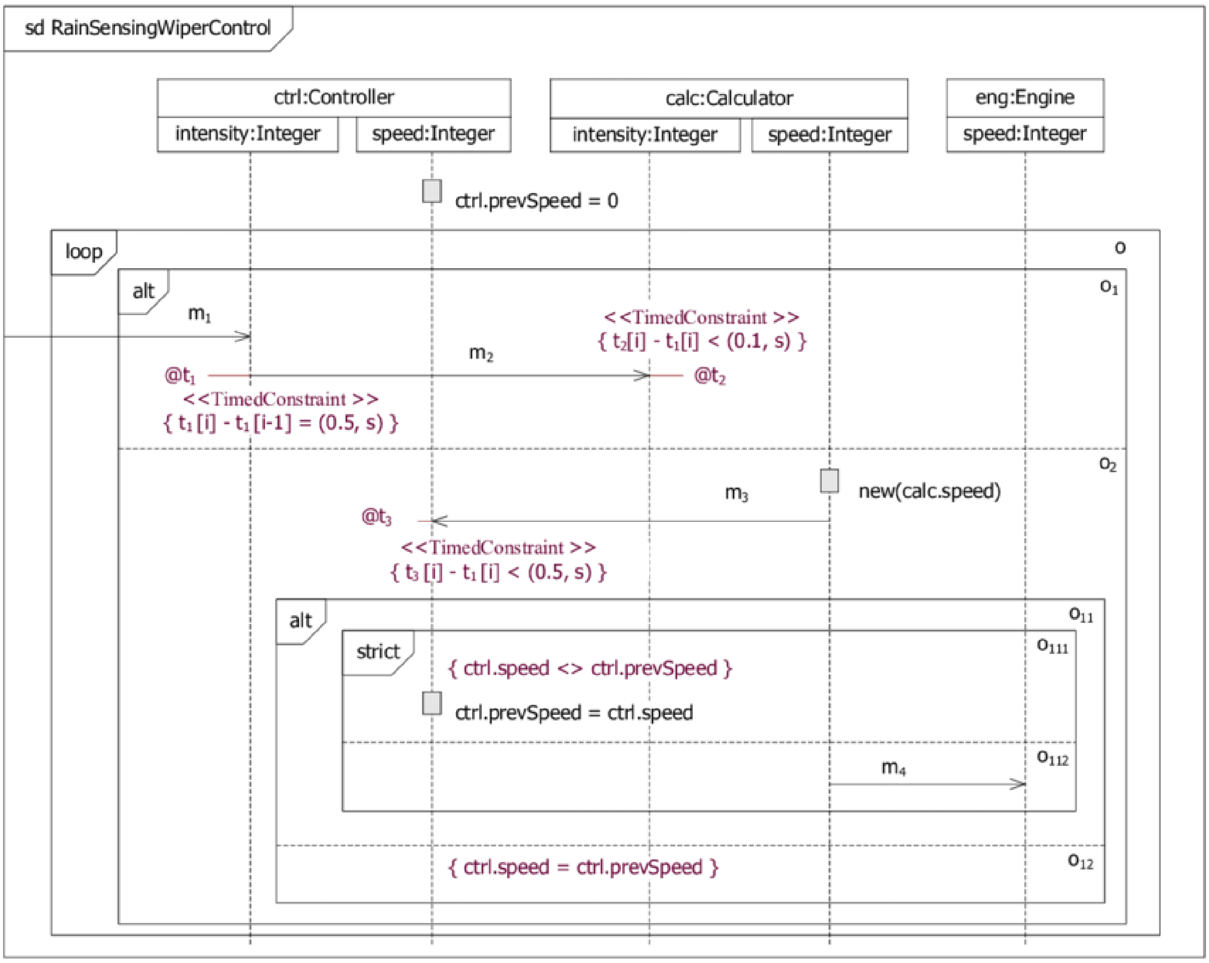


Figure 12 Block Diagram

Test Plan

High Level Test Plan

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_01	Manual Wiper System using STM32	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_02	Make the leds ON and OFF based on the user Input	User Input	SUCCESS	SUCCESS	Requirement Based
HLTP_03	Assigning the Interrupt Button	User Input	SUCCESS	SUCCESS	Requirement Based

ID	Description	Expected I/P	Expected O/P	Actual O/P	Type Of Test
HLTP_04	Our project should meet the Expected Output	User Input	SUCCESS	SUCCESS	Requirement Base

Low Level Test Plan

ID	HLTP ID	Description	Expected I/P	Actual O/P	Type Of Test
LLTP_01	HLTP_01	Should need the proper Folder Structure	SUCCESS	SUCCESS	Requirement Based
LLTP_02	HLTP_02	User our code need to be readable for any one	SUCCESS	SUCCESS	Requirement Based
LLTP_03	HLTP_03	Using Snake_case	SUCCESS	SUCCESS	Requirement Based
LLTP_04	HLTP_04	Should have the proper Lab Setup	SUCCESS	SUCCESS	Requirement Based

Implementation and Summary

Git Link:

Link: https://github.com/GENESIS-2022/MasteringMCU-Team76.git

Individual Contribution and Highlights

1. Wiper System using C Programming
2. Source code management using GitHub

Role in Project Team

1. Programmer: Done Programming for Wiper System
2. Tester: Writing Testcases and testing the integrated code

Miniproject 7 – Range Rover Evoque Project[Team]

Modules

1. Automotive Systems
2. GitRequirements

Anti-Theft Security System Anti-theft units, which provide protection through the ignition system. Under the hood there is a computer that controls the operation of the engine.

There are few Anti-Theft Security System available are

1. Tracking systems.
2. Factory-installed car alarms.
3. Immobilizing devices.
4. VIN etching.

High Level Requirements

SNo.	TITLE	Module	Description
SYS_1	Requirement	User(Driver)	Owns The Key Fob whether to turn On /Off Engine.
SYS_2	Requirement	Key Fob	Consist of Button lock, unlock, Hazard, Low beam and rear Door in the car. Which is used to Awake CPU
SYS_3	Requirement	Central Processing Unit	Take control over the module which send data stream to RF. Once it receives data from RKE engine start.
SYS_4	Requirement	Radio Frequency Transmitter	Used to send Radio Frequency Wave.
SYS_5	Requirement	Data Stream	connection-oriented communication, a data stream is a sequence of digitally encoded coherent signals used to transmit or receive.
SYS_6	Requirement	Remote Keyless Entry (RKE)	The remote keyless system's receiver in the car captures the RF signal, extracts it and sends the data stream to the CPU.

Design

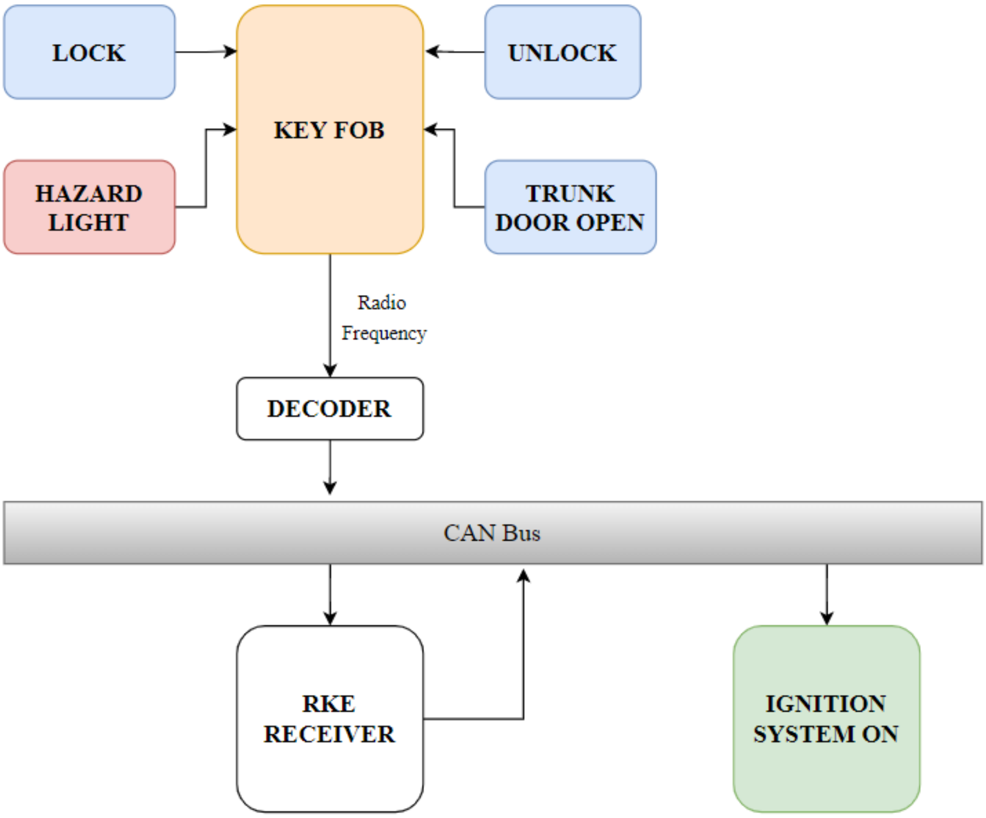


Figure 13 Structure Diagram

Implementation and Summary

Git Link:

Link: https://github.com/Ramki17/Automotive-System_RangeRover.git

Individual Contribution and Highlights

4. Antitheft System Case Study
5. Source code management using GitHub

Role in Project Team

1. Designer: Done Designing for Project
2. Researcher: Done case study for Antitheft System

Miniproject 8 – MINI AIRCRAFT[Team]

Modules

1. Matlab
2. Matlab Script

Requirements :

Introduction:

An electrical Energy management based on fixed priorities of the loads is considered a conventional implementation as applied in today’s aircraft systems. It can cut and reconnect loads depending on their importance. Further implementations are depicted that are able to eliminate certain drawbacks of such a typical load management

Objective:

The main objective of the project is to Reduce the enegy consumption of the Aircraft

Features:

This project supports the following types of Energy Management in Aircraft

1.Source Management:

If sources are available that can be connected in parallel one can apply a source management,that controls the different sources or generators in an energy efficient way.An intelligent source management will regulate the several sources to reach the overall power losses

2.Electrical Storage Device:

The degree of freedom of an energy management methodincreases considerably if electrical storage devices like bat-teries or supercaps are available. Storages can be used tosmooth out the power consumption of load groups. This in turnenables to design lighter generators, feeders, and convertersespecially in case of many non-constant loads. However, thebatteries or supercaps will add weight. Thus, there will bean optimal tradeoff between installed battery-capacity andinstalled power of e.g. generators to minimize weight

3.Exploit Slow Responding Roads:

In today's aircraft systems there is a number of slowresponding loads. That is systems and components with largetime constants like heaters. Since electrical storages will addweight, one can also try to decrease power peaks by exploitingsuch slow responding loads (SRL). Thus, they can be handledlike an electrical storage since they store energy in theirrespective physical state like the heat of a galley oven.

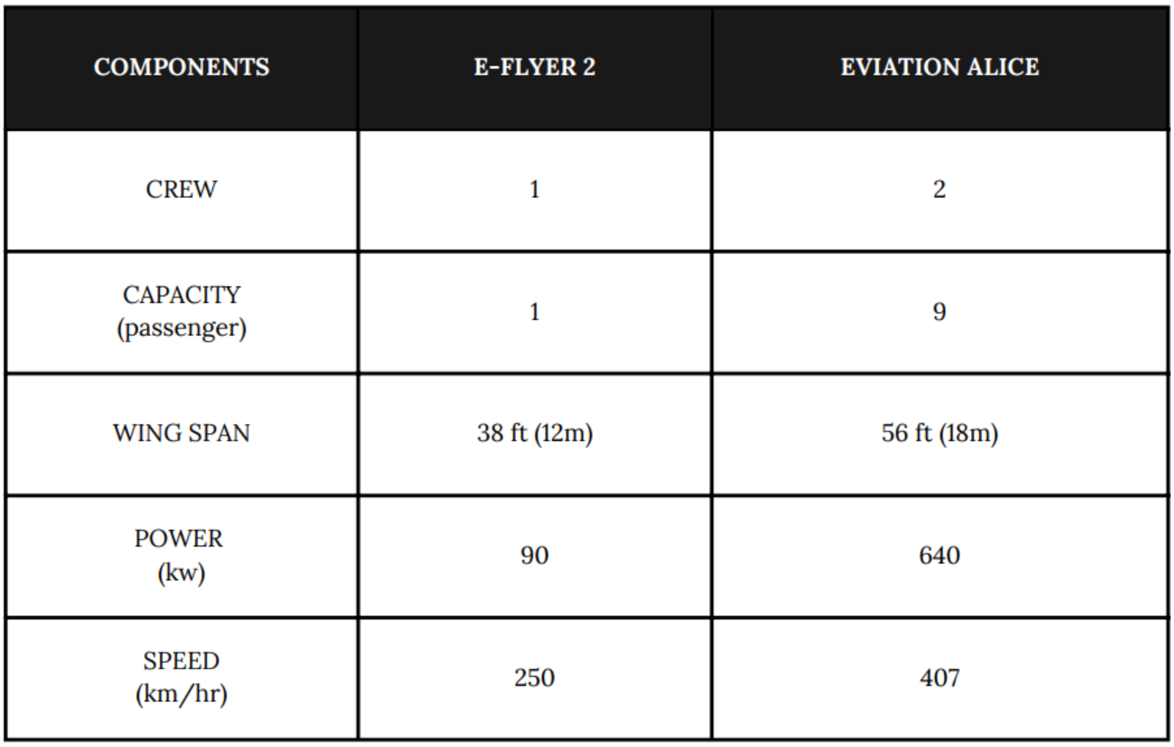
4.Variable Prorities:

To consider the changing importance of loads during a flightone can simply use variable priorities instead of fixed ones.Thus, the priority can be determined by the loads themselvesdepending on their current importance.

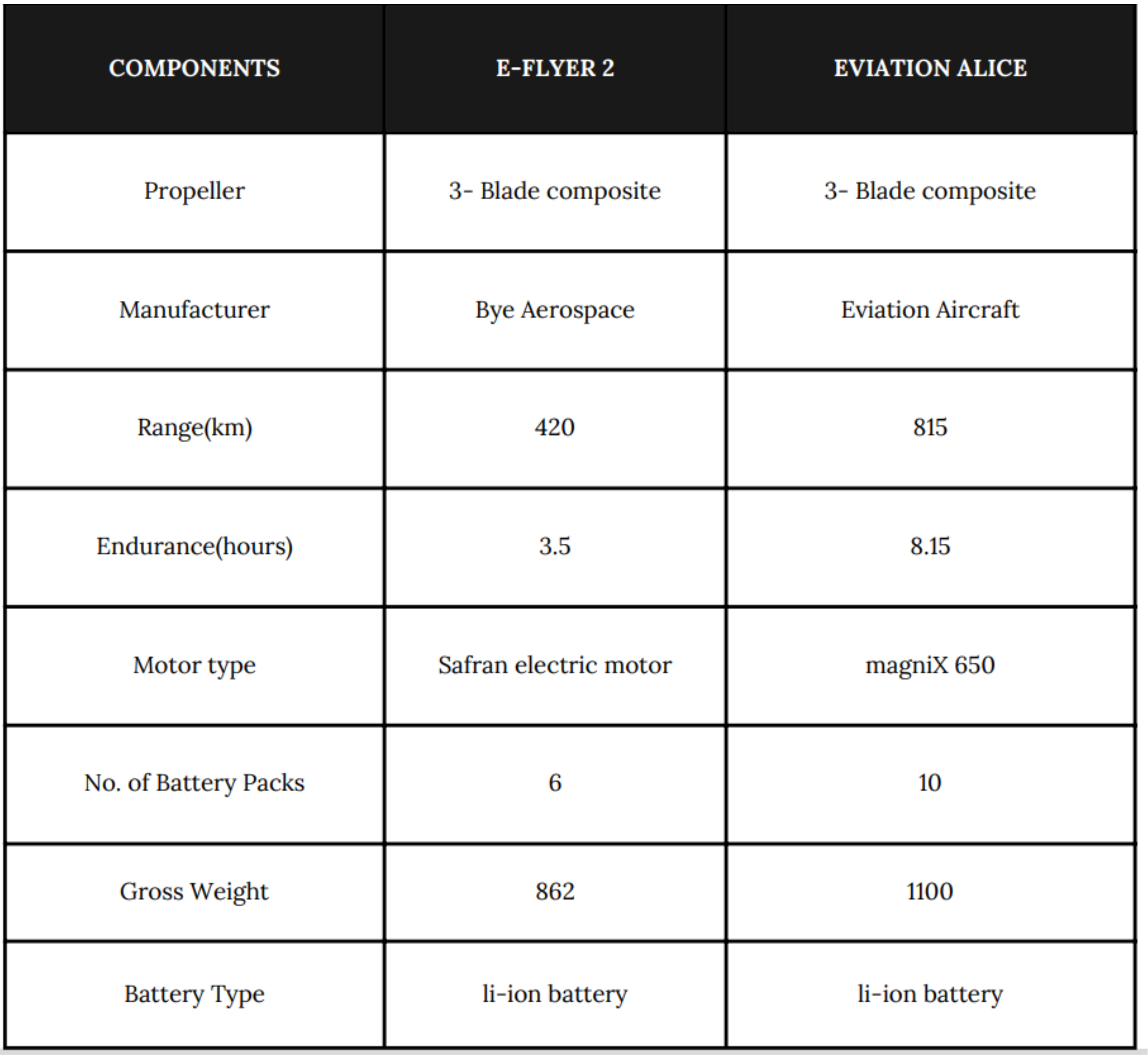
5.Supervise Reconnection

Instead of shedding loads if an overload occurs, one canalso prevent loads from being reconnected if a dedicatedpower level is reached.

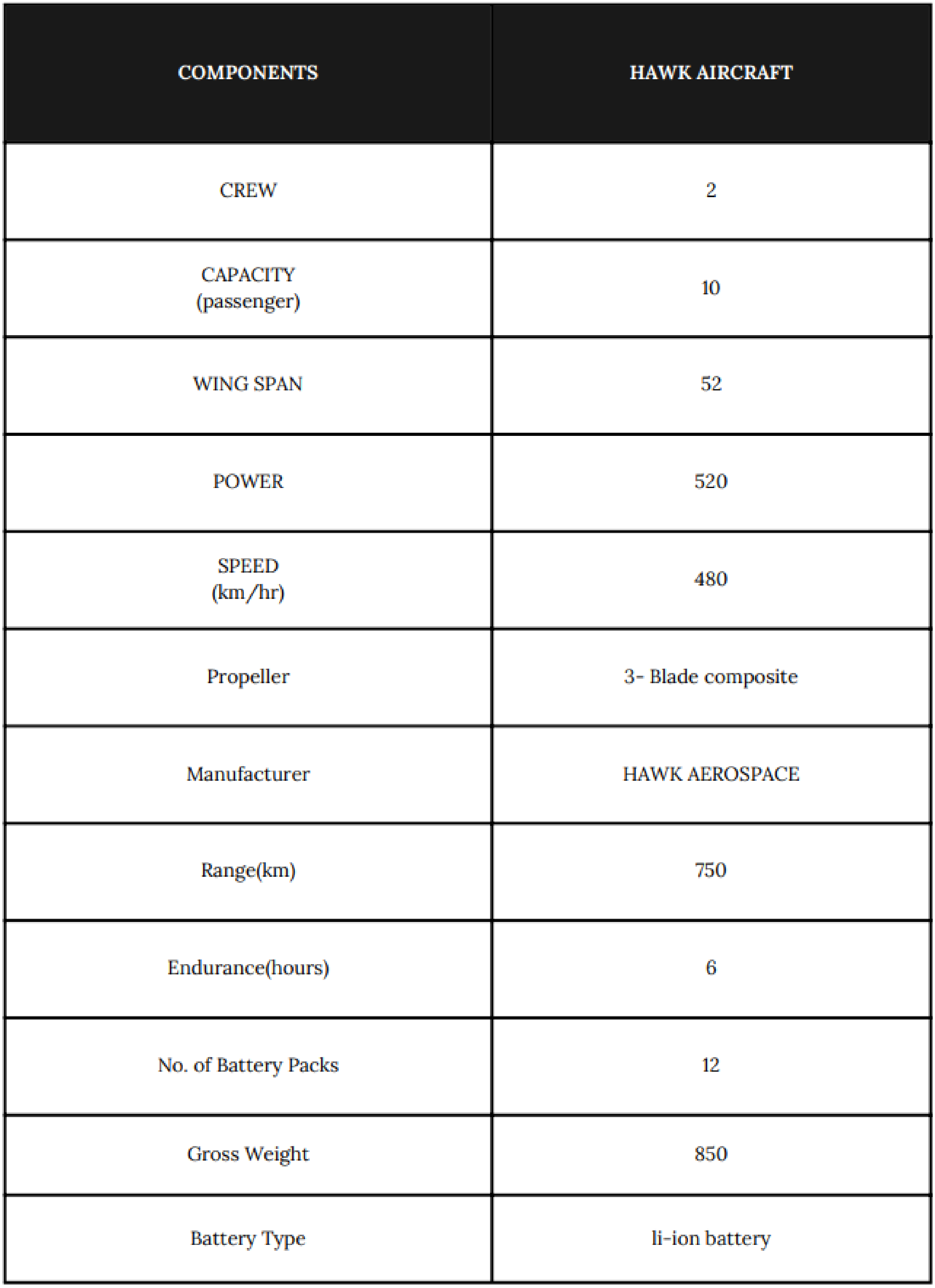
Eviation Alice:



COMPONENTS	E-FLYER 2	EVIATION ALICE
CREW	1	2
CAPACITY (passenger)	1	9
WING SPAN	38 ft (12m)	56 ft (18m)
POWER (kw)	90	640
SPEED (km/hr)	250	407



COMPONENTS	E-FLYER 2	EVIATION ALICE
Propeller	3- Blade composite	3- Blade composite
Manufacturer	Bye Aerospace	Eviation Aircraft
Range(km)	420	815
Endurance(hours)	3.5	8.15
Motor type	Safran electric motor	magniX 650
No. of Battery Packs	6	10
Gross Weight	862	1100
Battery Type	li-ion battery	li-ion battery



COMPONENTS	HAWK AIRCRAFT
CREW	2
CAPACITY (passenger)	10
WING SPAN	52
POWER	520
SPEED (km/hr)	480
Propeller	3- Blade composite
Manufacturer	HAWK AEROSPACE
Range(km)	750
Endurance(hours)	6
No. of Battery Packs	12
Gross Weight	850
Battery Type	li-ion battery

Simulation:

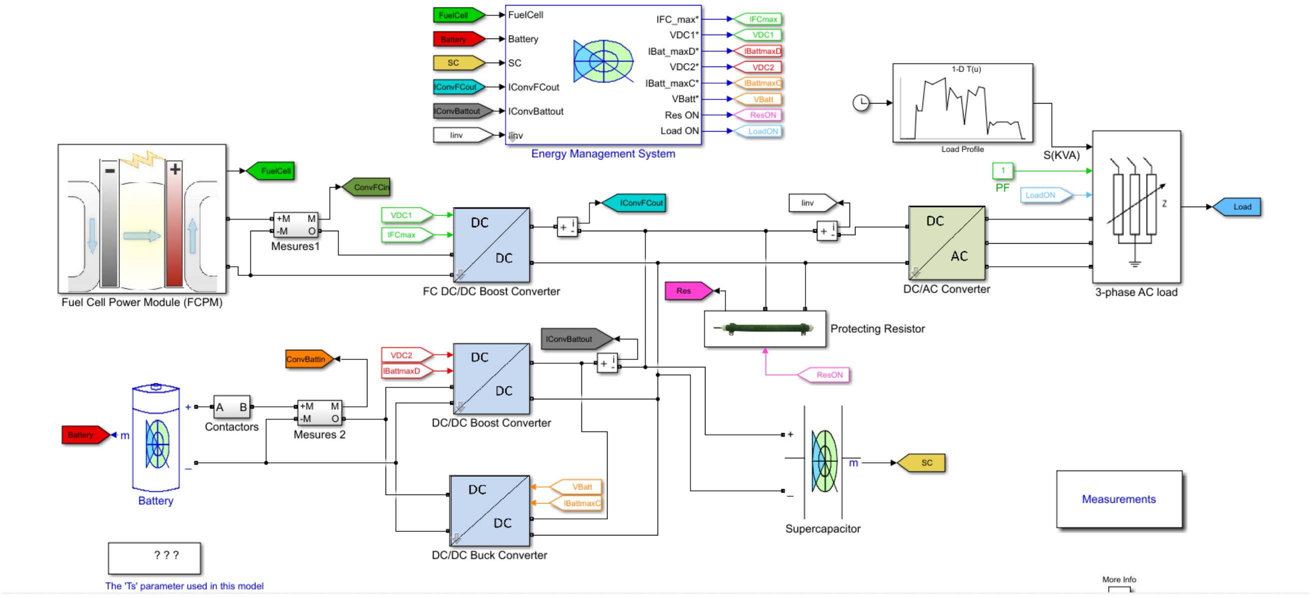


Figure 14Simulation diagram

Implementation and Summary

Submission: Submitted in GEALearn

Individual Contribution and Highlights

1. Done in Matlab Script

Role in Project Team

1. Done Matlab scripting for Mini Aircraft Bike
2. Researcher: Done case study for Mini Aircraft Bike

Miniproject 9 –Key Fob[Individual]

Modules

1. Autosar
2. Git

Requirements

Key Fob:

Consist of Button lock, unlock, Hazard, Low beam and rear Door in the car which is used to Awake CPU

SNo.	TITLE	Module	Description
SYS_1	Requirement	User(Driver)	Owns The Key Fob whether to turn On /Off Engine.
SYS_2	Requirement	Key Fob	Consist of Button lock, unlock, Hazard, Low beam and rear Door in the car. Which is used to Awake CPU

Design

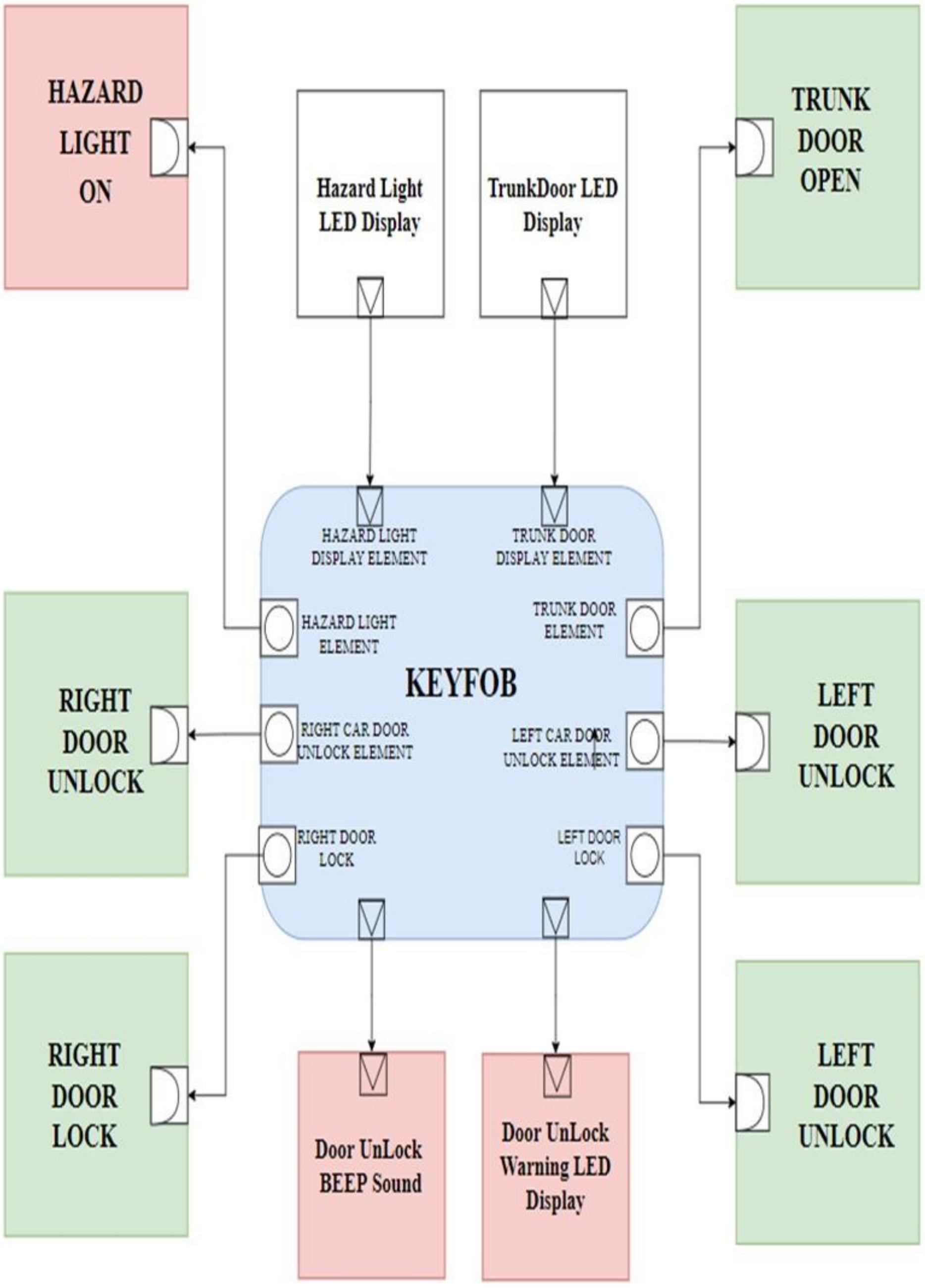


Figure 15 Uml Diagram

Implementation and Summary

Git Link:

Link: https://github.com/MidhunChakravarthi-06/KeyFob_40020506_DPS

Individual Contribution and Highlights

Key Fob Case Study

Source code management using GitHub

AtomicSwComponent

SWCInternalBehavior

SWCImplementation